SYBASE iAnywhere

SQL Anywhere and Microsoft .NET

A WHITEPAPER FROM SYBASE iANYWHERE

Contents:

Revision History

Version 1.1 - July 2010

Introduction

SQL Anywhere offers a number of integration features with the Microsoft .NET Framework. These features are designed to make it easier for developers to work with a SQL Anywhere database while developing a database-driven application. This whitepaper outlines the integrated Microsoft .NET support features of SQL Anywhere and includes a high-level description of the .NET Framework concepts. Each SQL Anywhere integration point is then explained and illustrated in more detail.

Topics include interaction between XML and relational databases, connecting to data sources using the SQL Anywhere ADO.NET and .NET Data Providers, using the ADO.NET Entity Framework with SQL Anywhere, managing website security using the SQL Anywhere ASP.NET Providers, writing .NET stored procedures and functions, and MobiLink synchronization .NET support.

A good understanding of the .NET concepts and these support features will help developers take advantage of full SQL Anywhere capabilites and achieve goals with less effort. While not compulsory, this paper assumes familiarity with the Microsoft .NET Framework and its components.

Understanding the .NET Framework

The .NET Framework is a programming model, which is designed to simplify application development in distributed environments. There are two main components to the .NET Framework:

- **Common Language Runtime**: A managed, protected application execution environment. The runtime manages code when it is executed and handles tasks such as memory and thread management. A virtual machine executes the supported .NET languages, including C#, Visual Basic, and C++.
- **Common Class Library**: A library of common classes that is available across all supported languages. Prior to .NET, there were different Windows APIs for each language.

The .NET Compact Framework from Microsoft is a subset of the .NET Framework for smart devices. It delivers managed code to smart devices, and also allows you to run secure, downloadable applications on Microsoft supported devices such as personal digital assistants (PDAs) and mobile phones.

When it comes to data management software such as SQL Anywhere, "supporting .NET" really means two things:

- Programmability using the .NET Framework and programming environments, and ADO.NET (the common database API in .NET) in particular.
- XML support. In particular, making data available in XML format and making functions available as Web services.

SQL Anywhere Integration Points with .NET

SQL Anywhere provides a wide range of support for .NET. Supported features include:

- SQL Anywhere allows you to export XML and store XML data in your database.
- SQL Anywhere allows you to expose your data via an XML web service.
- SQL Anywhere supports the ADO.NET interface using the SQL Anywhere .NET Data Provider, as well as the standard .NET OLE DB and ODBC Data Providers.
- SQL Anywhere allows you to use any supported .NET language (version 2.0) to write your stored procedures and functions.
- SQL Anywhere supports ADO.NET Entity Framework and provides integration components for Visual Studio that can be used to generate Entity Data Models from a SQL Anywhere database.
- SQL Anywhere ASP.NET Providers allow ASP.NET web developers to implement the authentication system and run the website on a SQL Anywhere database.
- SQL Anywhere is tightly integrated with Microsoft Visual Studio 2008 and 2010, makes it easier for developers to work with a SQL Anywhere database while developing an application.
- MobiLink exposes a .NET API for its synchronization client (DBMLSync).

The following sections outline these features in more detail, as well as explaining features and options that may be added to later releases of SQL Anywhere to enhance its .NET support.

XML and Relational Databases

Extensible Markup Language (XML) is a markup language for representing structured data in a text format. XML is designed to be a simple markup language, like HTML, but also to be flexible, like SGML. There is no specific format for data as XML, and the legal tag names and structure can be re-defined for each application. XML documents use a document type definition (DTD) or XML schema to define the structure, elements, and attributes that are used in an XML file.

.NET uses XML as a universal data-exchange format. The XML-based Simple Object Application Protocol (SOAP) is used by .NET as a mechanism for inter-process communication.

There are several ways in which XML and relational databases interact:

- Importing XML into relational databases
- Exporting XML from relational databases
- Storing XML documents in the database
- Performing XML queries
- Creating XML views
- Accessing Web services

Current releases of SQL Anywhere support exporting XML and storing XML documents in your SQL Anywhere database. There are also several procedures and functions provided for reading XML documents and importing their contents.

Importing XML Into Relational Databases

One way to import XML into your relational database is to parse the XML in an application and then generate SQL statements to insert, delete, or update tables. There are currently several parsers and tools available that allow you to do this, such as the built-in OpenXML procedures/functions.

The ADO.NET DataSet object allows you to read the data and/or schema from an XML document into a DataSet. The ReadXml method populates a DataSet from an XML document that contains both a schema and data, while the ReadXmlSchema method reads only the schema from an XML document. Once the DataSet is filled with the data from the XML document, you can update the tables in your database with the changes from the DataSet.

Both of these methods can be used with current releases of SQL Anywhere to import data from an XML document into your database.

Exporting XML from Relational Databases

XML can be exported directly in a SELECT statement, by using the FOR XML option.

SQL Anywhere also allows you to export XML from Interactive SQL. The OUTPUT statement supports an XML format that outputs query results to a generated XML file. The generated XML file is encoded in UTF-8 and contains an embedded DTD. In the XML file, binary values are encoded in character data (CDATA) blocks with the binary data rendered as 2-hex-digit strings.

There are two other ways you can export your data as XML in SQL Anywhere: use the ADO.NET DataSet object, or assemble the result set into an XML document within your application.

The ADO.NET DataSet object allows you to save the contents of the DataSet in an XML document. Once you have filled the DataSet (for example with the results of a query on your database) you can save either the schema or both the schema and data from the DataSet in an XML file. The WriteXml method saves both the schema and data in an XML file, while the WriteXMLSchema method saves only the schema in an XML file.

Alternatively, you can execute a SQL query and then assemble the results in an XML document in an application.

Storing XML in Relational Databases

SQL Anywhere supports two data types that can be used to store XML documents in your database: the XML data type and the LONG VARCHAR data type. Both of these data types store the XML document as a string in the database.

The XML data type uses the character set encoding of the database server. You can cast between the XML data type and any other data type that can be cast to or from a string. Note that there is no checking that the string is well-formed when it is cast to XML.

Querying XML

Once you have XML stored in the database, you may want to query it. This can be done using the built-in functions and stored procedures.

Please refer to the online documentation for more information about XML usage in SQL Anywhere:
http://dcx.sybase.com/index.html#1200en/dbusage/ug-sqlxml.html

XML Web Services

A Web service is programmable application logic that can be accessed by other applications in different locations via the Internet using a set of underlying standard Web protocols such as HTTP, Simple Object Access protocol (SOAP), and XML. Web services are designed to be platform and language independent.

Web Services Description Language (WSDL) is an XML-based language that is used to describe a Web service and how to access it. Universal Description Discovery and Integration (UDDI) is a registry for Web services; it's similar to a search engine, but for Web services. Neither WSDL nor UDDI are required for a Web service, but they make it easier for programs to find (using UDDI) and use (using the WSDL information) a Web service.

Web services in SQL Anywhere are supported via an integrated HTTP server and a SOAP request manager in the database server. This allows you to send SOAP requests to SQL Anywhere via HTTP, and then SQL Anywhere returns a response to the requesting HTTP client. Starting SQL Anywhere 12.0.0, HTTP services now support automatic connection pooling to maximize the effect of plan caching and benefit from the potential performance improvement.

Furthermore, there are several data formatting options available, which include the .NET data payload format. This formats the data in such a way that data-typing is preserved through transmission, and can be used properly from within your application.

Please refer to the online documentation for more information about web services in SQL Anywhere:
http://dcx.sybase.com/index.html#1200en/dbprogramming/http-web-services.html

ADO.NET and .NET Data Providers

ADO.NET is the latest data access API from Microsoft in the line of ODBC, DAO, RDO, OLE DB, and ADO. It is the preferred data access component for the Microsoft .NET Framework. .NET data providers are tools that provide access to data stores so that applications can retrieve and modify data from the data source. In ADO.NET, data providers (also called managed providers) are used to facilitate the integration of data with .NET applications.

The .NET Framework, as shipped by Microsoft, currently includes three data providers: the SQL Server .NET Data Provider, the OLE DB .NET Data Provider, and the ODBC .NET Data Provider. The SQL Server .NET Data Provider allows you to connect to Microsoft SQL Server version 7.0 or later databases, while the OLE DB .NET Data Provider allows you to access relational database systems (including SQL Anywhere), as well as data sources for which there is an OLE DB provider. The .NET Framework also supports Microsoft's ODBC .NET Data Provider for connecting to any ODBC data source.

The OLE DB driver included with SQL Anywhere .NET Data providers have been implemented to supplement the three already included in the framework. This allows you to develop applications in any of the supported .NET languages using the standard ADO.NET interface for all Windows platforms including Windows Mobile.

Alternatively, the standard non-managed SQL Anywhere Data Provider can be accessed via the OLE DB, and ODBC data providers.

The ADO.NET Programming Interface

Each managed provider implements the following classes, which are the standard objects for database manipulation and management:

- **Connection**: connects to a data source
- **Command**: executes a command (SQL statements or stored procedure) against a data source
- **DataReader**: provides forward only, read-only access to results of a command
- **DataAdapter**: fills a DataSet and handles updates to data
- **Parameters**: passes parameters to a Command object
- **Transaction**: provides COMMIT and ROLLBACK functionality
- **Error, Exception**: handles and collects error and/or warning messages

One of the primary objects in ADO.NET is the DataSet. The DataSet is a disconnected store for data retrieved from a database. It is a collection of DataTable, DataRow, DataColumn, and DataRelation objects. The DataSet is a generic object provided by Microsoft as part of the ADO.NET architecture. Using a provider's DataAdapter, you can fill a DataSet, modify the data in the DataSet, and then apply the changes to the database through the DataAdapter. The DataSet is independent of any managed provider or database driver and can be used to read and write either data or schema information in XML.

Caution should be used when updating your database from a DataSet. Any changes you make to a DataSet are made while you are disconnected. This means that your application does not have locks on these rows in the database. Your application must be designed to resolve any conflicts that may occur when changes from a DataSet are applied to the database.

MobiLink provides a solution for resolving conflicts in a replication environment. MobiLink is a session-based relational database synchronization system that is intended for two-way synchronization of data between a central, consolidated database and a large number of remote databases. It allows you to choose selected portions of data for synchronization and includes features that allow you to resolve conflicts between changes made to data in different remote databases.

The SQL Anywhere .NET Data Provider is a native .NET data provider for SQL Anywhere databases. Unlike the other supported providers, it communicates directly with SQL Anywhere and does not require bridge technology. It also runs on Windows Mobile.

The .NET data provider implements the iAnywhere.Data.SQLAnywhere namespace and allows you to write programs in any of the .NET supported languages, including C# and Visual Basic .NET, and access data from any SQL Anywhere database.

The following are some key features of the SQL Anywhere .NET Data Provider:

- **Support for .NET languages**: The SQL Anywhere .NET Data Provider can be used with all .NET supported languages, including C# and Visual Basic .NET.
- **Application flexibility:** The SQL Anywhere .NET Data Provider supports all classes, including DataReader and DataAdapter.
- **Support for .NET languages**: The SQL Anywhere .NET Data Provider can be used with all .NET supported languages, including C# and Visual Basic .NET.

- **Support for connection pooling**: The SQL Anywhere .NET Data Provider supports connection pooling, which allows your application to reuse existing connections rather than repeatedly creating a new connection to the database. This may improve performance.
- **Superior performance**: The SQL Anywhere .NET Data Provider is a direct implementation, and does not require OLE DB or ODBC bridges. It is faster than the .NET OLE DB Data Provider.
- **Broad Microsoft Windows support**: The SQL Anywhere .NET Data Provider supports both Windows and Windows Mobile platforms. Currently, Microsoft does not supply an OLE DB or ODBC managed provider that allows you to access data on Windows Mobile. The SQL Anywhere .NET Data Provider is the only way you can access SQL Anywhere using the .NET Compact Framework on Windows Mobile.

ADO.NET Entity Framework

The ADO.NET Entity Framework is a new technology from Microsoft .NET Framework 3.5 SP1 that allows developers to bridge the gap between relational data representations and application business logic. Databases are typically designed by database administrators (DBAs) who ensure that all information is stored in schemas optimized for data retrieval. However, these schemas, while efficient for data access, are not the best way to represent the business objects that applications require. Often times, programmers must be ready to design and implement creative methods for transforming the data stored in relational systems into robust and reusable objects that are integral parts of enterprise solutions. The Entity Framework presents software developers with an intuitive approach to design business logic based on relational data.

The Entity Data Model (EDM) is a conceptual view of the data used in the application. Often, existing database schemas contain large amounts of data, and only a subset of this data is required by client programs. For example, a Microsoft Windows client application that processes online orders may only require information about customers, products, and orders. Instead of including all the database tables in the EDM, the developer provides only the tables that are relevant to the application.

Continuing with its tradition of supporting the latest data access technologies, SQL Anywhere allows developers to take advantage of the ADO.NET Entity Framework for building powerful database-driven applications. The SQL Anywhere integration tools with Visual Studio also enable developers to create an entity data model from a SQL Anywhere database and view the visual representation of the model appears in the Entity Designer. Developers can then access the data stored in the database using the Language Integrated Query (LINQ) to Entities, Object Services, and the EntityClient Provider methodologies. The entity classes are partial class entities, which means that developers are also able to extend the class entities to include properties and methods that complete business objects and provide desired business logic.

The SQL Anywhere ASP.NET Providers

The successor to Microsoft's Active Server Pages (ASP) technology, ASP.NET, is a web application framework that allows website developers to build dynamic web sites, web applications and web services. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework also allows ASP.NET components to process SOAP messages. In addition Visual Studio offers a rich, integrated web development environment as well as access to powerful libraries and methods, makes designing and implementing fully functional websites even easier.

Implementing the authentication and authorization systems in web applications used to be a complicated task for the developers. Thankfully, starting with ASP.NET 2.0, web developers no longer need to write and re-write the code to store and validate credentials, create and manage users, monitor the status and performances of websites etc. Instead, the ASP.NET providers provides secure and extensible implementations for managing security related information in the web applications. Developers can simply configure the providers and administrate the web site by using the provided ASP.NET Website Administration Tool.

The SQL Anywhere ASP.NET providers allow you to build a security mechanism using a SQL Anywhere database as backend storage. They must be installed in order to add the required schema to a designated database for storing and managing confidential user information. A setup wizard is also provided to setup the schema to any SQL Anywhere databases automatically with only a few configurations required.

SQL Anywhere includes five providers:

- **Membership Provider:** provides authentication and authorization services.
- **Roles Provider:** provides methods for creating roles, adding users to roles, and deleting roles. Use the roles provider to assign users to groups and manage permissions.
- **Profiles Provider:** provides methods for reading, storing, and retrieving user information such as user preferences.
- **Web Parts Personalization Provider:** provides methods for loading and storing the personalized content and layout of web pages.
- **Health Monitoring Provider:** provides methods for monitoring the status of deployed web applications.

Please refer to the online documentation for more information about SQL Anywhere ASP.NET Providers: http://dcx.sybase.com/index.html#1200en/dbprogramming/aspdotnet-providers.html

CLR Stored Procedures and Functions

SQL Anywhere includes support for CLR stored procedures and functions. A CLR stored procedure or function allows you to reference a function or procedure written in C#, VB.NET, or any other .NET language that resides in a class library (.dll). The database will load the library, and call the function in place of a SQL stored procedure. This allows you to manipulate data in any .NET language, thus preserving and re-using your business logic.

Please refer to the online documentation for more information about the CLR external environment in SQL Anywhere: http://dcx.sybase.com/index.html#1200en/dbprogramming/pg-extenv-clr.html

MobiLink .NET support

MobiLink is a session based synchronization system that allows two-way synchronization between a main database, called the consolidated database, and many remote databases.

MobiLink Synchronization Scripts

Synchronization scripts are used to control the actions of the MobiLink synchronization server. Generally, the synchronization scripts are written as stored procedures in the SQL language of the consolidated database, or in Java.

SQL Anywhere also allows you to use any of the supported .NET programming languages to write MobiLink synchronization scripts. This gives you full access to all the functionality of the .NET Common Language Runtime, and you can write scripts in C#, Visual Basic .NET, or any other supported .NET language.

Using .NET synchronization logic allows you to perform operations across database platforms, and provides portability across RDBMSs. With .NET synchronization logic, you can use MobiLink to access data from application servers, Web servers, and files. You can use iAnywhere classes in your synchronization logic to access data on the consolidated database's synchronization connection. For example, you can write a .NET script to use an external server to validate a user ID and password in the server's authenticate_user event.

Scripts also allow you to access and manipulate uploaded data in the consolidated database before it is committed. For example, you could reject a change before it is committed so that other remotes would not receive it. If you use an external program to access the data on the consolidated database, you cannot view or undo the update until it has been committed.

This is also supported for use within UltraLite databases. UltraLite is a smaller footprint alternative to SQL Anywhere used to build and deploy relational database applications on small devices.

Direct Row API

Alternatively, the Direct Row API passes data directly through the .NET environment, allowing you to manipulate the raw synchronized data before committing it to the database.

MobiLink .NET API for DBMLSync

DBMLSync is the synchronization client that initiates client-side synchronization. The Dbmlsync .NET API, using the iAnywhere.MobiLink.Clinet namespace, provides a programming interface that allows MobiLink client applications written in C++ or .NET to launch synchronizations and receive feedback about the progress of the synchronizations they request. This enables you to access a lot more information about synchronization results and integrate synchronization seamlessly into your applications.

Please refer to the online documentation for more information about MobiLink and synchronization scripts: http://dcx.sybase.com/index.html#1200en/mlserver/writing-scripts-synch.html

Summary

The SQL Anywhere .NET Data Provider provides native access to your .NET applications, including applications running on Windows Mobile. In addition, you can currently import, export, and store XML in your SQL Anywhere database.

SQL Anywhere supports the ADO.NET Entity Framework technology and provides the integration components for Visual Studio to generate the Entity Data Model from a SQL Anywhere database. You can use Integrated Query (LINQ) to Entities, Object Services, and the EntityClient Provider methodologies to access data stored inside a SQL Anywhere database.

The SQL Anywhere ASP.NET Providers allow you to run your website on a SQL Anywhere database by providing secure and extensible implementations for managing roles, membership, and profiles in your web applications.

MobiLink gives you full access to the functionality of the .NET Common Runtime Languages by allowing you to write your synchronization logic in any of the supported .NET languages.

Additional Resources

For additional resources such as whitepapers, tutorials, and sample code, please visit the SQL Anywhere .NET Development Center available at this location:
http://www.sybase.com/developer/library/sql-anywhere-techcorner/microsoft-net

Related Topics

SQL Anywhere Integration with Visual Studio 2010:
http://www.sybase.com/detail?id=1080131

SQL Anywhere and the ADO.NET Entity Framework:
http://www.sybase.com/detail?id=1060541

AdventureWorks2008 .NET Samples for SQL Anywhere:
http://www.sybase.com/detail?id=1061456

Tutorial: Creating an ASP.NET Web Page using SQL Anywhere:
http://www.sybase.com/detail?id=1080238

Connecting to a SQL Anywhere Database Using ADO.NET and the SQL Anywhere .NET Data Provider:
http://www.sybase.com/detail?id=1054947

SYBASE, INC.
WORLDWIDE HEADQUARTERS
ONE SYBASE DRIVE
DUBLIN, CA 94568-7902 USA
Tel: 1 800 8 SYBASE

www.sybase.com/ianywhere

Copyright © 2009 Sybase, Inc. All rights reserved. Unpublished rights reserved under U.S. copyright laws. Sybase, and the Sybase logo are trademarks of Sybase, Inc. or its subsidiaries. All other trademarks are the property of their respective owners. ® indicates registration in the United States. Specifications are subject to change without notice. 3/09.